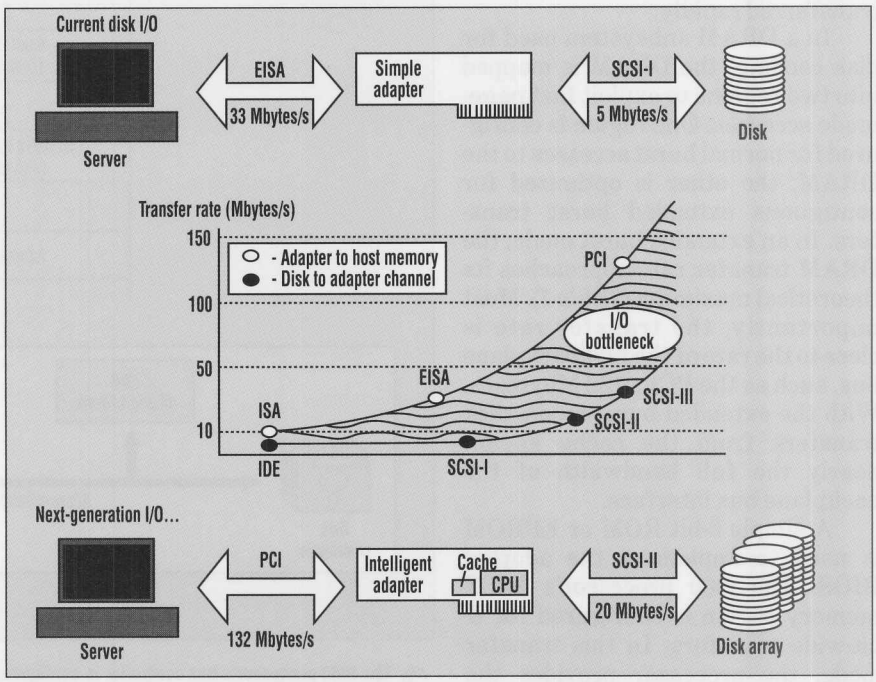
Reprinted from ELECTRONIC DESIGN December 1993

Breaking the I/O Bottleneck

• DAVE DANNENBERG • *Intel Corp., 5000 W. Chandler Blvd., Chandler, AZ 85226; (602) 554-2429.*

Growing numbers of desktop PCs are connected to file and application servers in client/server configurations, in an environment in which PCs, networks, and servers form a web of distributed and tightly coupled computing power. This type of architecture relies on the smooth flow of large amounts of data to make it work. As communications are improved in one part of the system, bottlenecks are likely to appear elsewhere. For example, in today's systems, a data bottleneck is emerging at the desktop and server network connection, but an even tighter bottleneck has appeared in the PC server's disk I/O.

Data transferred from a PC server typically traverses two sequential paths before reaching the server's host memory: a disk-to-adapter channel, such as the Small Computer System Interface (SCSI); and the adapter-to-host backplane bus, such as EISA or the faster Peripheral Communication Interface (PCI). As backplane bus speeds in the PC servers leap to the speeds of PCI, the disk-to-adapter channel begins to limit performance. (*Fig. 1*)



1 While backplane bus speeds have increased from the ISA's 10-Mbytes/s to almost 150 Mbytes/s with the PCI, the disk-to-adapter channel hasn't been able to keep pace, creating an I/O bottleneck.

High performance 32-bit microprocessors, such as Intel's i960 processor, have the intelligence and data-handling agility to relieve the disk I/O bottleneck. The processor possesses facilities that attack three separate facets of the problem:

- Multichannel communication, as implemented in Redundant Array of Inexpensive Disks (RAID)
- Large, fast disk caches at the adapter, bypassing the disk-to-adapter data paths
- Large, fast disk caches at the adapter, bypassing the disk-to-adapter channel altogether for many accesses.

The external bus design of the i960 processor optimizes memory and peripheral data transfers, making it an optimal data-transfer engine for high-bandwidth SCSI traffic and concurrent transfer across a backplane bus, such as PCI. The processor's external bus design provides the capability for highly tuned interfaces to external memory and peripherals, without incurring the cost and complexity typical in designing with 32-bit microprocessors. The processor also uses a programmable bus-interface unit that interfaces to moderate to slow-speed peripherals as well as high-speed DRAM with little or no interface logic.

The i960 processor's bus controller is configured differently for interfaces with each of the four external subsystems that comprise the intelligent disk adapter (*Fig. 2*):

- The main memory subsystem, for disk cache, program code, and program data.
- The disk adapter BIOS used to initialize the adapter.
- One or more disk controllers, such as SCSI controllers.
- The interface to the host memory, such as a PCI backplane bus.

The bus interface is configured according to memory region (or range of memory addresses). Each memory region is programmed to match the characteristics of the external memory subsystem.

The main memory subsystem is the most complex component of the server adapter design. DRAM is economically the best choice to implement the large cache memory for the server adapter. Larger caches translate directly to higher I/O data rates from the server. Caching controllers often support up to 64 Mbytes of on-card disk cache. With implementation of large caches, cached accesses may be limited to the speed of the DRAM subsystem.

To optimize the DRAM interface, access-interleaving techniques can be used to obtain the highest performance. The most efficient DRAM interface requires external logic to generate row and column address strobes, as well as logic to multiplex the address. In the case of an inter-leaved design, external

logic is required to demultiplex the data path. The high-performance DRAM sub-systems take advantage of fast page-access capabilities of DRAM. In this access mode, blocks of contiguous data can be transferred rapidly.

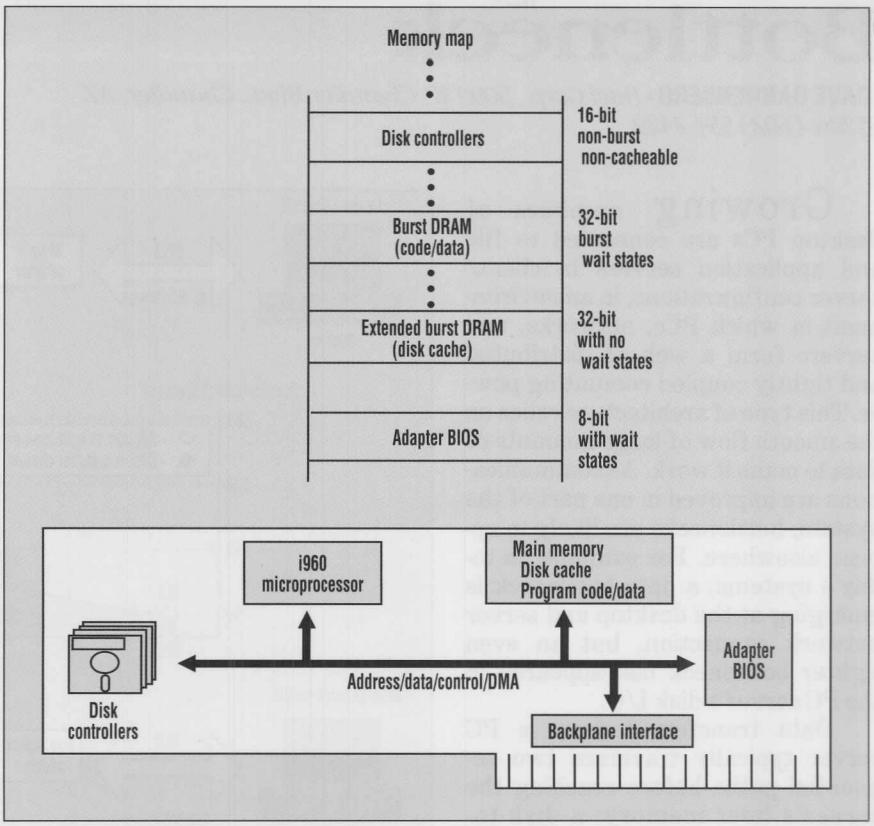
In a DRAM subsystem used for disk caching, the DRAM is mapped into two regions to exploit fast page-mode accesses. One region is configured for normal burst accesses to the DRAM; the other is optimized for contiguous extended burst transfers. In an extended burst mode, the DRAM transfer rate approaches its theoretical maximum (*Table 1*). Most importantly, the transfer rate is close to the rate of the fast backplane bus, such as the PCI's 132 Mbytes/s. With the extended burst mode, disk transfers from the cache exploit nearly the full bandwidth of the backplane bus interface.

A simple 8-bit ROM or EPROM is used to implement the adapter BIOS. The i960 processor's BIOS memory region is configured for 8-bit-wide transfers. In this transfer mode, the processor provides the lower two-byte address lines. The logic that's used to decode the low address bits externally isn't needed. The BIOS memory region is programmed for simple non-burst accesses consistent with ROM reads. The programming eliminates external state machines that otherwise "unburst" the burst access for the ROM's benefit. In addition, the BIOS region is set up with delay cycles, referred to as wait states. The wait states accommodate the slow access characteristics of ROMs. Again, the programming eliminates external logic that would be required for memory wait-state control.

Similarly, the region for the disk-controller devices is configured to match the peripheral device characteristics. Besides the characteristics mentioned for the BIOS interface, the disk-controller region can be set as non-cacheable to ensure coherency with the i960 processor's on-chip data cache. This eliminates external logic that otherwise is needed to detect non-cacheable memory.

The i960 processor's on-chip DMA controller serves as an ideal interface to the disk controllers. The DMA controller provides an autonomous control over these devices. The DMA interface is suited for the adapter's multitasking environments to handle simultaneous caching operations or host memory transfers concurrently with data transfers from the disk.

The server adapter's final component is the backplane interface. The PCI bus closely resembles an i960 processor burst-mode bus. The bridge logic to the PCI can be implemented relatively simply. The backplane interface is probably implemented as a secondary bus master from the i960 processor's perspective. In this way, the interface can use the i960's low-latency bus-master capabilities to perform direct transfers from cache to host memory. Alternatively, the i960 can implement these directly by using fly-by DMA transfers, using the backplane interface as the source or destination port for DMA transfers. ED



2 The I960 processor's bus controller is configured differently as an interface for each of the four external subsystems: main memory, adapter BIOS, backplane interface, and the disk controllers.

TRANSFER BANDWIDTHS FROM MAIN MEMORY

	Transfer bandwidth (Mbytes/s @ 33-MHz system bus)		
Bus configuration	70-ns DRAM (non-interleaved)	70-ns DRAM (2-way interleaved)	20-ns SRAM
Burst mode	48.5	66.7	133
Extended burst mode	59	107	N/A